Artificial Intelligence Group project

Solving a Faculty's Timetable Scheduling Problem using Genetic Algorithms

Prepared by

Adham Ali El-Din Assem 202000101

Ziad Khaled Abd El-Fattah 202000352

Abdelrahman Khaled Mohamed 2020000513

Kirollos Rafik Rateb 202000679

Youssef Emiel Habashi 202001079

Youssef Ramy Shawky 202001088

Youssef Salah Amin 202001091

1. Introduction

**1.1 Purpose**

Create a schedule for classes and a timetable to avoid scheduling conflicts.

**1.2 Intended Audience**

College students, or any student who wants to have the most convenient schedule, are the intended audience for this software.

**1.3 Intended Use**

Determine the most optimum schedule.

**2. Overall Description**

Given a set of lecturers, a set of courses on individual topics, and a Course Requirements matrix with integer elements representing the number of hours a lecturer teaches a course during each week, the challenge is to assign timings to these hours such that a student can enroll in as many relevant course combinations as is practical.

**2.1 Assumptions and Dependencies**

Create a workable schedule for the each department of the faculty that allows courses from various departments to be combined in a variety of ways to suit the needs of individual students. This schedule must ensure that there are no scheduling conflicts, brief gaps between lectures, and fewer days spent in the faculty.

**3. Main functionality**

In universities, the semester timetable scheduling process has become very complex since there are many student groups and some of the student groups follow some of the courses together. The problem is to allocate times to the number of hours a lecturer teaches a course during each week so that a student may take as many suitable combinations of courses as possible.

Courses should be assigned to specific time slots for five working days of the week taking into consideration the specific classrooms suitable for the respective courses and the registered number of students.

**In creating a timetable an institute has to face many constraints which could be defined as:**

**3.1 Hard Constraints** (should be satisfied no matter what to come up with a feasible solution):

• No two courses can be scheduled at the same time for the same department.

• No two courses can be scheduled at the same time in a particular lecture hall.

• A lecturer cannot be teaching two courses at the same time.

• The lecture hall should have the capacity to hold the number of students following the scheduled course.

**3.2 Soft Constraints** (not that essential to satisfy better to satisfy as much as possible):

• Minimum number of gap hours between lectures.

The problems associated with hard constraints need to be resolved to produce a functional solution. To optimize the performance of the scheduling it is important to address the issues linked to soft constraints. The aim of this work is to get the usefulness of genetic algorithm to obtain optimal solutions in general timetable scheduling.

**4. Similar applications in the market**

**4.1 Single bus line timetable optimization :** A case study in Beijing

Bus lines are suffering from serious decline in passenger volume due to the rapid development of urban rail transit and shared transport, and big data intelligence may help them change the [status quo](https://www.sciencedirect.com/topics/mathematics/status-quo). However, the tremendous amount of travel data collected in recent years have not got effectively utilization. In order to improve passenger volume for bus lines, this paper devotes to develop a data-driven bus timetable to substitute the existing experience-based bus timetable, which is now widely used by bus lines. Driven by the bus GPS data and IC card data, a timetable optimization model with time-dependent passenger demand and travel time among stops is proposed. The objective of maximizing passenger volume is based on a new preference-based passenger selection model. The working hours constraint is initially formulated, and the headway constraint and departure time constraints are also taken into account. For handling the step functions in both objective and constraints, we introduce a set of 0–1 variables to transform the proposed model into an [integer linear programming](https://www.sciencedirect.com/topics/computer-science/integer-linear-programming). A model contraction approach is provided for solving the medium-scale problems and a two-stage solution method is proposed for the large-scale problems. The proposed model and methodology are tested on a real-world bus line in Beijing. The results show that it is able to produce a satisfactory timetable that outperforms the previously used experience-based one in terms of raising the average passenger volume by 8.2%.

**4.2 Faculty's Timetable Scheduling Problem**

weekly timetable for classes in a college for a class. We must arrange classes and come up with a timetable so that there are no clashes between classes. Here, the task is to search for the optimum timetable schedule. A possible definition for the problem is: Given a set of lecturers, a set of courses on individual topics and a Course Requirements matrix with integer elements representing the number of hours a lecturer teaches a course during each week, the problem is to allocate times to these hours so that a student may take as many suitable combinations of courses as possible. Or, simply to create a practical timetable for a whole faculty in which courses offered by different departments may be combined in various ways to suit individual students.

**4.3 Optimum workforce scheduling under the (14,21) days-off timetable**

An efficient optimum solution is presented for a real life employee days-off scheduling problem with a three week cycle. Over a given work cycle, each worker is given 14 successive workdays and 7 successive off days. Given different labor demands for each day of the week, the primary objective is to minimize the number of workers. The secondary objective is to reduce transportation cost by minimizing the number of active days-off patterns.

**5. Academic publications (papers) relevant to the idea**

**5.1** Genetic Algorithm for University Course Timetabling Problem

From University of Mississippi

Link: [Genetic Algorithm For University Course Timetabling Problem.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EbgXqV9ruJ1BlKMV7iEIgx4BR93GgJ16hzM7eZoFe5s0VA?e=dBms7w)

**5.2** University Timetable Scheduling Using Genetic Algorithm

Approach Case Study: Rajarata University OF Sri Lanka

Link: [University\_Timetable\_Scheduling\_Using\_Ge.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EQcaIXhs7dpPvTK2Y-0pjNcBbBep-OOoxoejlgXUStW2OA?e=KpOdO0)

**5.3** Maastricht University Timetable Scheduling for the Department

of Data Science and Knowledge Engineering

Link: [MRP\_Report\_Timetabling\_at\_DKE.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EcKzdi_12pRIlgeJyJOL6p0Bklgiha0kyY3l2i3iKNOp1A?e=EgQzYH)

**5.4** Developing Course Timetable System For Academic

Department Using Genetic Algorithm

Link: [2-DOI10.5455-jjcit.71-1465735040.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EZEEkDrKhRxDvOz3xpXvbn0Bu3rsPlz7Yt8FxcduHptfpA?e=4Q0GTE)

**5.5** Exam Timetabling Problem Using Genetic Algorithm

Engineering Department, Ibra College of Technology, Sultanate

of Oman

Link: [IJRET20140305120.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EYQTCn8vZO9EtnWqsLGNUHoB0KLs3mC3qWKYDxZMAZ167A?e=oxzUsR)

**5.6** Timetable Scheduling System Implementing Genetic Algorithm

by O. RAMAREA L. RADIKGOMO O. SIMULAN. MOSELE

Link: [TIMETABLE\_SCHEDULING\_SYSTEM\_IMPLEMENTING.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EeRsKgNsLy1Fuy0jdULAeboBbYmGWSAL2kh-qxGJxFfWpQ?e=SPOZZm)

**5.7** Timetable Scheduling via Genetic Algorithm Final Year Thesis

2019. National University of Ireland, Galway Computer Science

and Information Technology

Link: [Timetable-Scheduling-via-Genetic-Algorithm-Andrew-Reid-East.pdf](https://fcihelwanedu-my.sharepoint.com/:b:/g/personal/kyrollos_rafiq_2017_fci_helwan_edu_eg/EYOMFcapxYhGoZn-3ANcaAUBlUMQkLW6l6s4OAYQYqZzUA?e=xMUkQy)

**6. Details of the algorithm(s)/approach(es) that will be used**

Genetic algorithm (GA) is a powerful problem-solving programming technique. It is in a category of evolutionary algorithms which is a subset of evolutionary computation in artificial intelligence. It closely mimics the biological model of chromosomes and genes with each chromosome representing an individual organism and genes forming components of a solution that is to be used with a genetic algorithm.

**How Genetic Algorithms Work**

**6.1 Generating an initial population of Timetables**

A genetic algorithm creates an initial population of 10 Timetables with each Timetable (chromosome) representing a complete solution to a given problem. The genes which constitute the Timetables are initialized to random values.

**6.2 Evaluating the suitability of each Timetable that forms the population**

fitness function is the form of a numeric score based on the number of conflicts between lectures and gap hours between them to evaluate the solutions (Timetables). In nature there is no assignment of a score the -- organisms just die or survive.

**6.3** **Selecting the Timetables for mating based on the above results**

If there are only two members among thousands available as parents, the variety of the offspring will be very much limited, which could stunt the process of evolution. The obvious conclusion is to make a large mating pool to obtain better results. Using as example:

> Roulette Wheel Selection.

> Selection based on the wheel of fortune.

> Tournament selection.

**6.4 Producing offspring by cross over the selected Timetables**

creation of an improved population is the aim of mating the most suitable Timetables in a population. It results in the production of offspring has better genes of the past generation by stepping over each class and by choosing a random number the function decides from which table to assign the class to the new generation.

**6.5 Mutating genes randomly**

Additional variety to the offspring DNA which was created during the crossover by changing a random value of specific class as class time or lecturer. A mutation may produce a desirable or undesirable character. Natural selection will decide upon the fate of the mutated Timetable.

**6.6 Ending the algorithm**

Terminating the program when one of the Timetables reaches a fitness value of 1 which means there is no conflicts on lecturers appointments, lectures, or gap hours. If no tables reach this value the algorithm will terminate after 500 generation.

**7. Diagrams**

**7.1 Flowchart**

Diagram

Description automatically generated

**7.2 Block diagram**

Diagram

Description automatically generated

**8. Development platform**

**8.1 Tools**

Pycharm IDE

**8.2 Programming language**

Python 3.8

**8.3 Libraries**

Prettytable: For printing the timetables in readable way as table

Random: For generating random numbers

**Code + documentation link:**

[kirollosR/TimeTable-Scheduling-using-genetic-algorithm (github.com)](https://github.com/kirollosR/TimeTable-Scheduling-using-genetic-algorithm)